AMENDMENTS TO THE CLAIMS

Please amend the claims as indicated below. The language being added is underlined ("__") and the language being deleted contains either a strikethrough ("——") or is enclosed by double brackets ("[[ ]]").

LISTING OF CLAIMS

1. (Previously Presented) A multi-pass method of rendering a plurality of graphic primitives comprising:

in a first pass:

passing only a limited portion of graphic data for each primitive through a graphic pipeline, wherein the limited portion of graphic data comprises location-related data;

processing the limited portion of graphic data to build a compressed z-buffer, the compressed z-buffer comprising a plurality of z-records, each z-record embodying z information for a plurality of pixels such that condensed depth information for the plurality of pixels is represented by a single z-record;

setting a visibility indicator, for each primitive, if any pixel of the primitive is determined to be visible;

in a second pass:

for each primitive, determining whether the associated visibility indicator for that primitive is set;

discarding, without passing through the graphic pipeline, the primitives for which the associated visibility indicator is not set;

passing the remaining portion of graphic data for each primitive determined to have the associated visibility indictor set;

performing a two-level z-test on graphic data, wherein a first level of the z-test compares the graphic data of a current primitive with corresponding information in the compressed z-buffer, and wherein a second level of the z-test is performed on a per-pixel basis in a z-test manner, wherein the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines that some but not all pixels of an associated macropixel are visible; and

communicating data associated with pixels of macropixels determined to be visible to a pixel shader for rendering.

2. (Canceled)

3. (Previously presented) The method of claim 1, wherein location-related data comprises X, Y, Z, and W values.

4. (Original) The method of claim 1, wherein each compressed z-record comprises a minimum z value for the plurality of pixels, a maximum z value for the

plurality of pixels, and a coverage mask, the coverage mask indicating which of the plurality of pixels are visible for the current primitive.

5. (Original) The method of claim 1, wherein each compressed z-record comprises two minimum z values for the plurality of pixels, two maximum z values for the plurality of pixels, and a coverage mask, the coverage mask indicating which of the plurality of pixels are visible for the current primitive.

6. (Original) The method of claim 1, wherein setting the visibility indicator more specifically comprises setting a bit in a frame buffer memory.

7. (Original) The method of claim 1, wherein the discarding is performed by a parser.

8. (Currently Amended) A method of rendering a plurality of graphic primitives comprising:

passing, within a graphic pipeline, only a limited portion of the graphic data associated with each primitive, wherein the limited portion of graphic data comprises location-related data; and wherein each primitive comprises a plurality of pixels;

processing the limited portion of graphic data associated with each individual primitive to build a compressed z-buffer for each primitive, wherein each compressed z-buffer contains a plurality of z-records which each contains compressed z-information for a macro-pixel;

creating a visibility mask for each primitive, wherein for each primitive, the visibility mask indicates whether the primitive is clipped, culled, or is a zero-pixel primitive;

determining, for each primitive, whether the primitive has at least one visible pixel based on the visibility mask;

communicating data associated with pixels of primitives determined to have at least one visible primitive to a pixel shader for rendering; and

passing and processing, within the pixel shader, the remaining graphic data associated with each primitive only for those primitives determined to have at least one visible pixel, wherein the remaining graphic data includes at least one of the following: lighting, texture, and fog data.

9. (Original) The method of claim 8, further comprising setting a visibility indicator for each pixel determined to have at least one visible pixel.

10. (Original) The method of claim 9, wherein setting the visibility indicator more specifically comprises setting a bit in a frame buffer memory.

11. (Canceled)

12. (Original) The method of claim 8, wherein the determining whether the primitive has at least one visible pixel ensures that the primitive does not fail a compressed z-buffer test, ensures that all pixels of the primitive are not culled, ensures that the primitive does not render to zero pixels, and ensures that all pixels of the primitive are not clipped.

13. (Previously Presented) A method of rendering a plurality of graphic primitives comprising:

passing in a first pass, within a graphic pipeline, only a limited portion of graphic data for each primitive, wherein each primitive comprises a plurality of pixels and wherein the limited portion of graphic data comprises location-related data;

processing the limited portion of graphic data to build a compressed z-buffer, the compressed z-buffer comprising a plurality of z-records, each z-record embodying z information for a plurality of pixels such that condensed depth information for the plurality of pixels is represented by a single z-record;

in a second pass, within the graphic pipeline, performing a two-level z-test on graphic data, wherein a first level of the z-test compares the graphic data of a current

primitive with corresponding information in the compressed z-buffer, and wherein a second level of the z-test is performed on a per-pixel basis in a z-test manner, wherein the second level z-test is performed only on pixels within a record of the compressed z-information in which the first level z-test determines that some but not all pixels of a macropixel are visible, wherein additional graphic data associated with each primitive is passed into the graphics pipeline on the second pass only for primitives that are at least partially visible; and

communicating data associated with pixels of macropixels determined to be visible to a pixel shader for rendering.

14. (Currently Amended) A graphics processor comprising:

first-pass logic configured to deliver to a graphic pipeline, in a first pass, only a limited portion of graphic data for each primitive, wherein each primitive comprises a plurality of pixels, wherein the limited portion of graphic data comprises location-related data;

logic configured to process the limited portion of graphic data for each primitive to create a compressed z-buffer comprising a plurality of z-records, wherein z-information for a macro-block is compressed into each of the plurality of z-records such that condensed depth information for the macro-block is represented by a single z-record;

logic configured to create a visibility mask for each primitive, wherein for each primitive, the visibility mask indicates whether the primitive is clipped, culled, or is a zero-pixel primitive;

logic configured to determine, for each primitive, whether the primitive has at least one visible pixel based on the visibility mask; and

second-pass logic configured to deliver to the graphic pipeline, in a second pass, the remaining graphic data associated with each primitive for only those primitives determined to have at least one visible pixel, the second-pass logic further configured to inhibit the delivery of graphic data to the graphic pipeline for primitives not determined to have at least one visible pixel.

15. (Original) The graphics processor of claim 14, wherein the first-pass logic and second-pass logic are contained within a parser.

16. (Original) The graphics processor of claim 14, wherein the logic configured to determine whether the primitive has at least one visible pixel ensures that the primitive does not fail a compressed z-buffer test, ensures that all pixels of the primitive are not culled, ensures that the primitive does not render to zero pixels, and ensures that all pixels of the primitive are not clipped.

17. (Original) The graphics processor of claim 14, further including logic for setting a visibility indicator for each primitive determined to have at least one visible pixel.

18. (Original) The graphics processor of claim 17, wherein the visibility indicator includes a single bit in a frame-buffer memory.

19. (Original) The graphics processor of claim 17, further including logic configured to associate each primitive processed in the first pass of the data with a distinct visibility indicator.

20. (Original) The graphics processor of claim 19, further including logic configured to evaluate, for each primitive presented for processing in the second pass, a status of the visibility indicator associated with the given primitive.

21. (Currently Amended) A graphics processor comprising:

logic configured to pass and process only a portion of graphic data passed into a graphic pipeline for each of a plurality of primitives, in a first pass within the graphic pipeline to determine whether the primitive has at least one visible pixel, wherein each primitive comprises a plurality of pixels, and wherein the limited portion of graphic data comprises location-related data;

logic configured to build a compressed z-buffer from processing of the graphic data in the first pass, the z-buffer comprising a plurality of z-records, wherein z-information for a macro-block is compressed into a single record such that condensed depth information for the macro-block is represented by a single record;

logic configured to create a visibility mask for each primitive, wherein for each primitive, the visibility mask indicates whether the primitive is clipped, culled, or is a zero-pixel primitive; and

logic configured to render, in a second pass within the graphic pipeline, only the primitives determined in the first pass to have at least one visible pixel based on the visibility mask, wherein the remaining portion of graphic data associated with each primitive is passed into the graphics pipeline on the second pass.

22. (Original) The graphics processor of claim 21, wherein the logic configured to limit the processing ensures that the primitive does not fail a compressed z-buffer test, ensures that all pixels of the primitive are not culled, ensures that the primitive does not render to zero pixels, and ensures that all pixels of the primitive are not clipped.

23. (Original) The graphics processor of claim 21, wherein the logic configured to limit the processing of graphic data is within a parser.

24. (Canceled).

25. (Original) The graphics processor of claim 21, further including logic for setting a visibility indicator for each primitive processed in the first pass.

26. (Original) The graphics processor of claim 21, further including logic configured to evaluate the visibility indicator for each primitive prior to submitting the primitive to the logic configured to render in the second pass.

27. (Previously Presented) The graphics processor of claim 26, wherein each compressed z-record comprises two minimum z values for the plurality of pixels, two maximum z values for the plurality of pixels, and a coverage mask, the coverage mask indicating which of the plurality of pixels are visible for the current primitive.